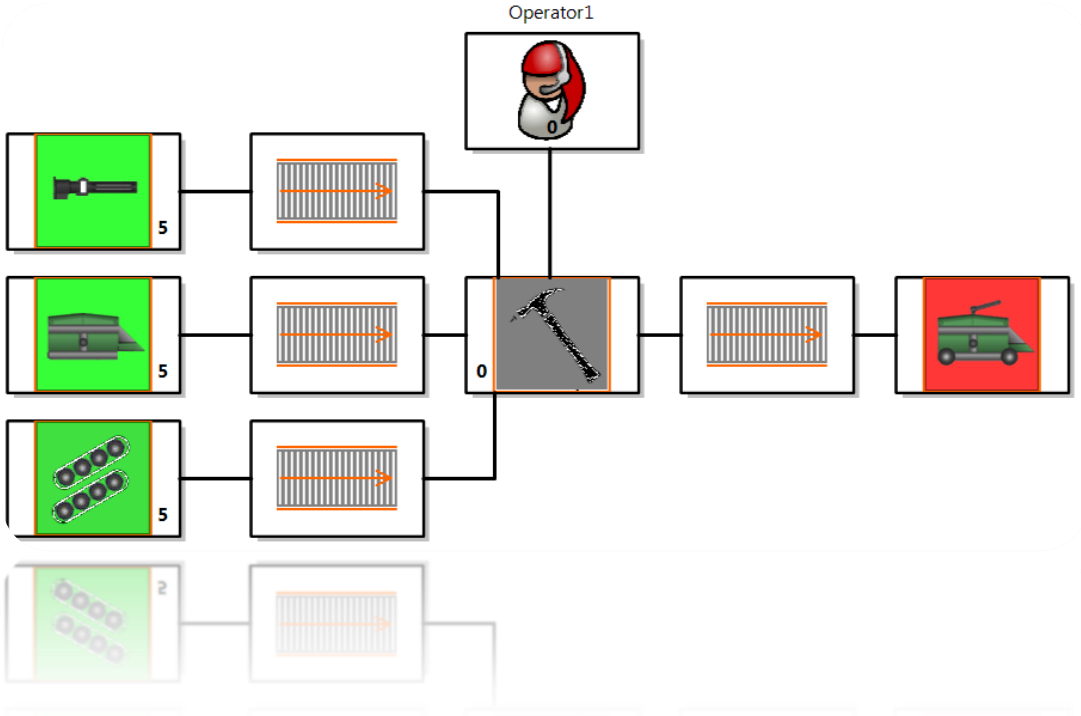
Comparison of Visual Studio's VMSDK1 and AToM3

(Matthias De Cock)



Overview

- Introduction
- (Meta-)modeling in VMSDK
- Validation
- Simulation
- Comparison
- Conclusion
- Questions

Introduction

- Two tools under consideration
  - Visualization and Modeling SDK[1]

[1] VMSDK, April 2010
archive.msdn.microsoft.com/vsvmsdk

Introduction

- Two tools under consideration
  - Visualization and Modeling SDK[1]
    - Steep learning curve
    - Difficult to find documentation
    - Support for simulation?

[1] VMSDK, April 2010
archive.msdn.microsoft.com/vsvmsdk

Introduction

- Two tools under consideration
  - Visualization and Modeling SDK[1]
    - Steep learning curve
    - Difficult to find documentation
    - Support for simulation?
  - AToM3[2]

[1] VMSDK, April 2010
archive.msdn.microsoft.com/vsvmsdk
[2] AToM3, January 2008
http://atom3.cs.mcgill.ca/

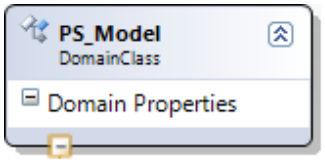
Introduction

- Two tools under consideration
- APC Production system

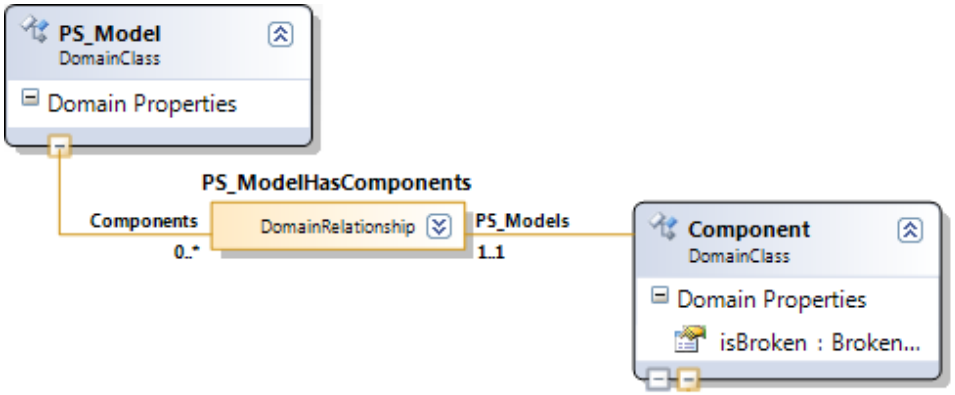
Introduction

- Two tools under consideration
- APC Production system
  - Constraints
    - Machine can have at most one output
    - No self-loops with conveyor belts
    - Generators/Garages
    - Cardinalities

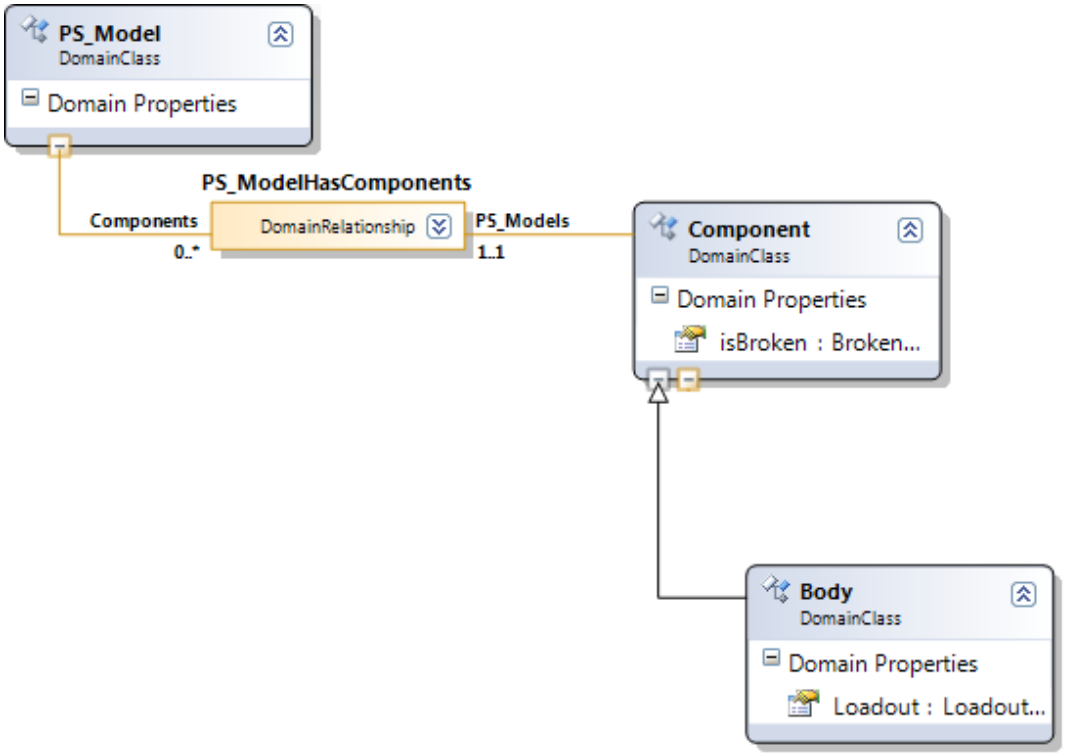
(Meta-)modeling in VMSDK



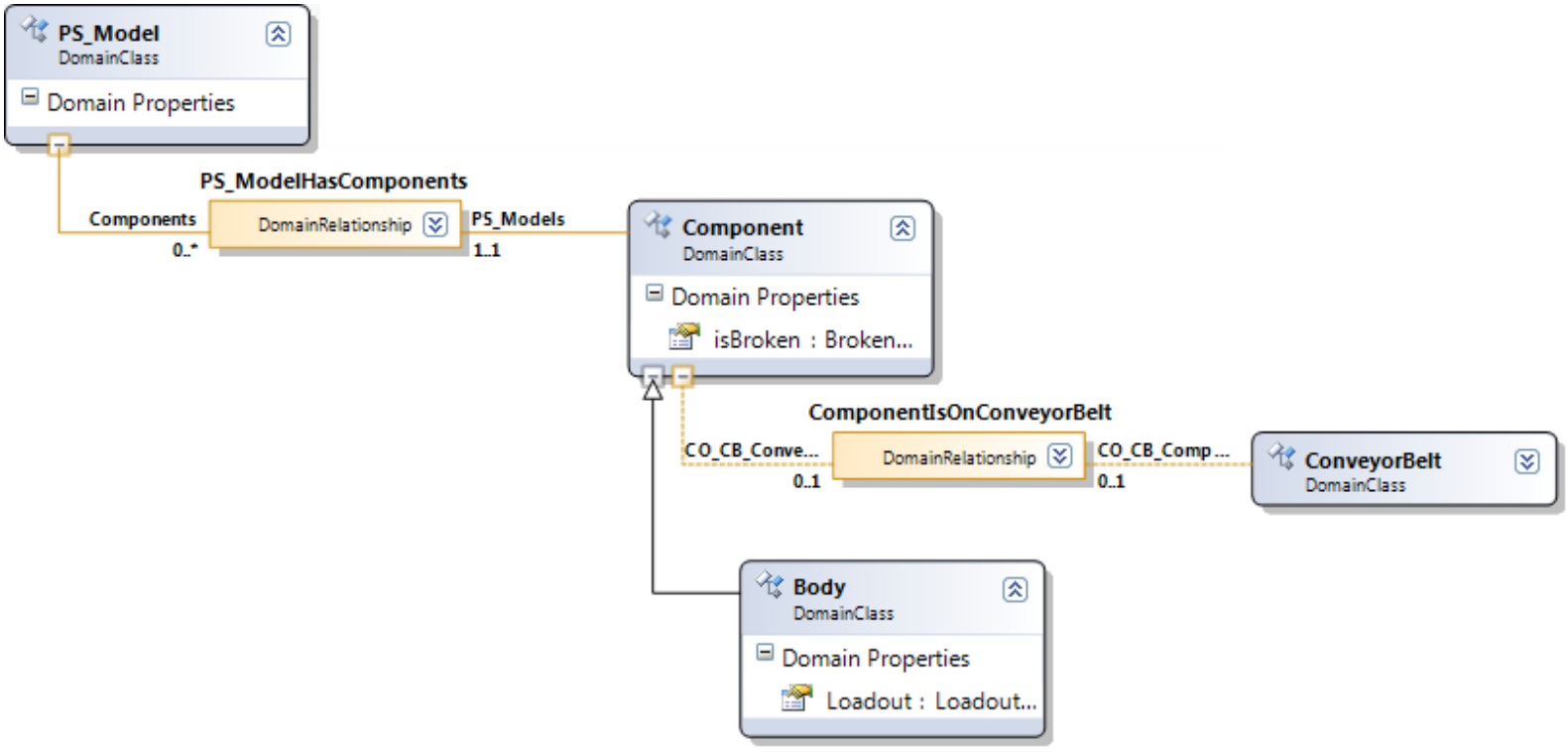
(Meta-)modeling in VMSDK



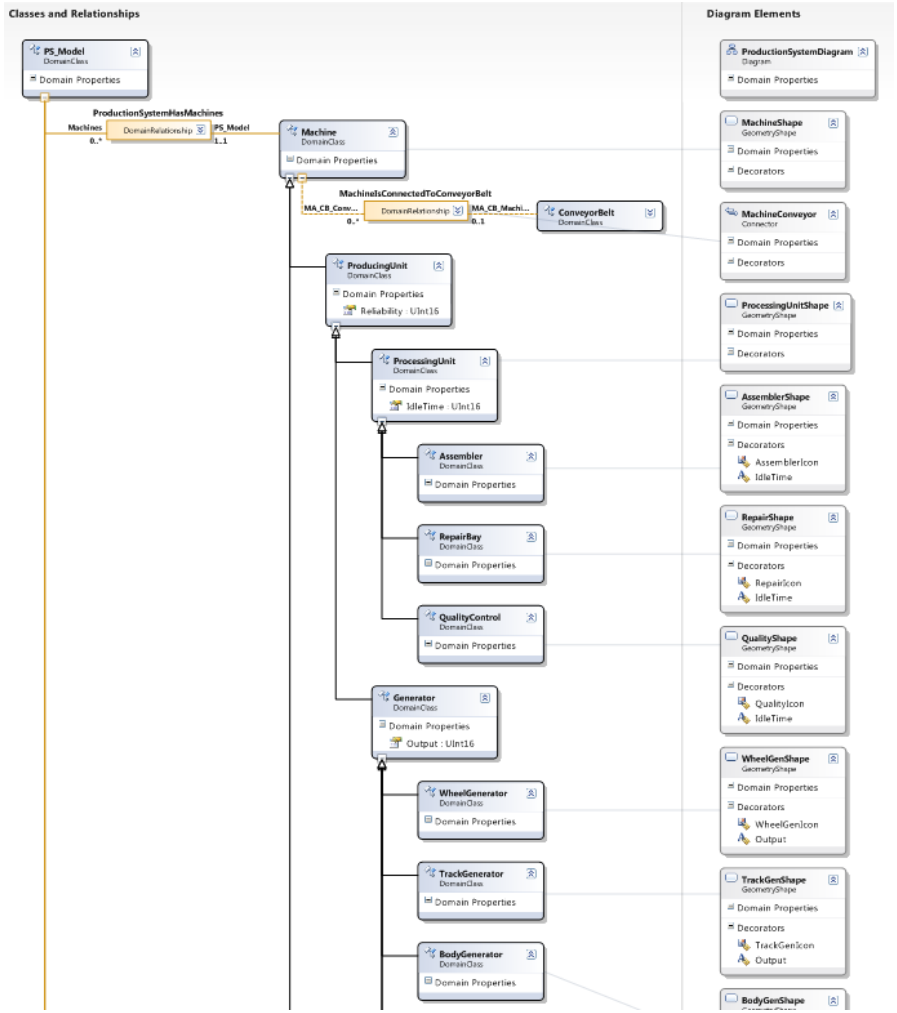
(Meta-)modeling in VMSDK



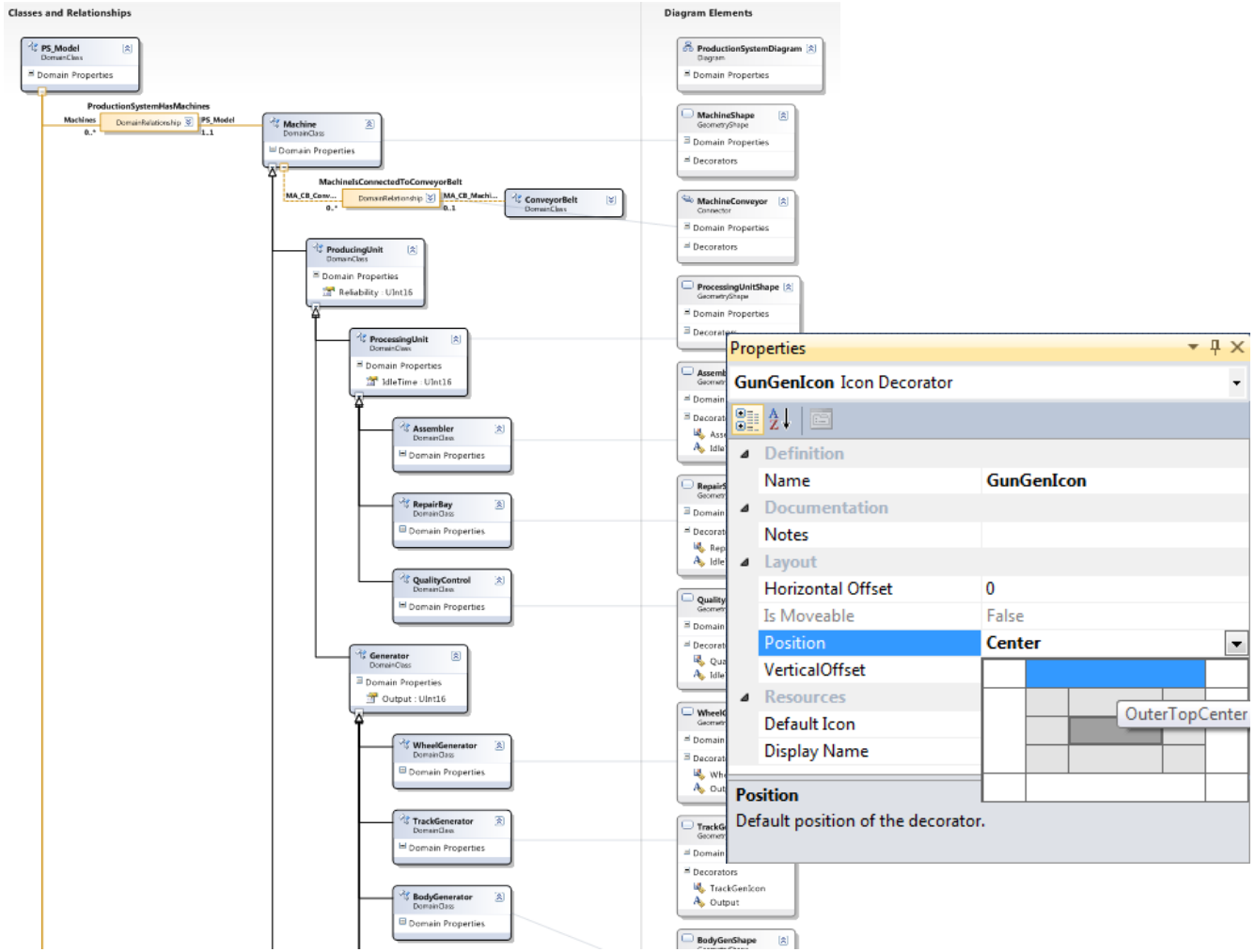
(Meta-)modeling in VMSDK



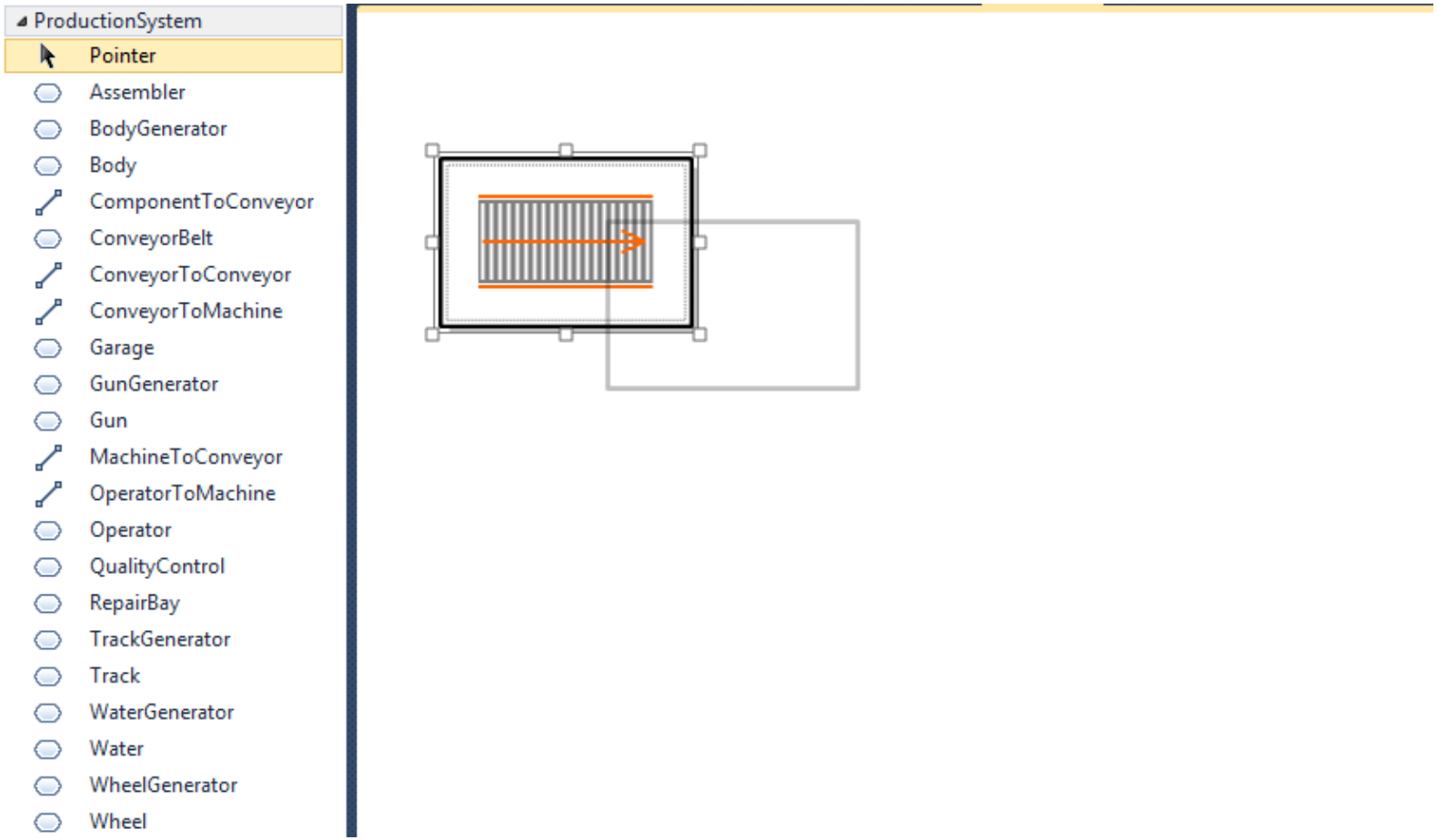
(Meta-)modeling in VMSDK



(Meta-)modeling in VMSDK



(Meta-)modeling in VMSDK

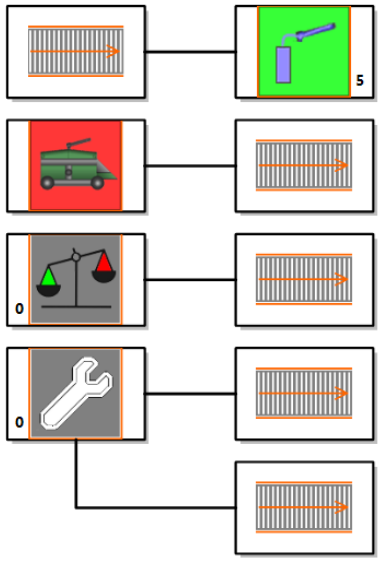


Validation

- Three types of validation

Validation

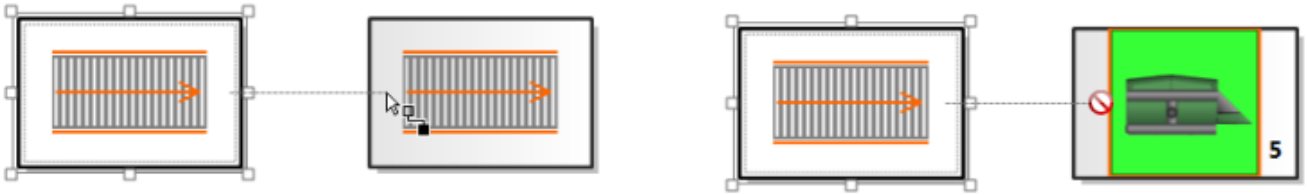
- Three types of validation
  - Validation on request



	Description	File
3	A Quality Control station should have exactly two outgoing links	Validation.pr
4	A Garage should not have any outgoing links	Validation.pr
5	A Generator should not have any incoming links	Validation.pr
6	A Machine (with exception of the Quality Control) should have only 1 outgoing link	Validation.pr

Validation

- Three types of validation
  - Validation on request
  - Interactive constraints

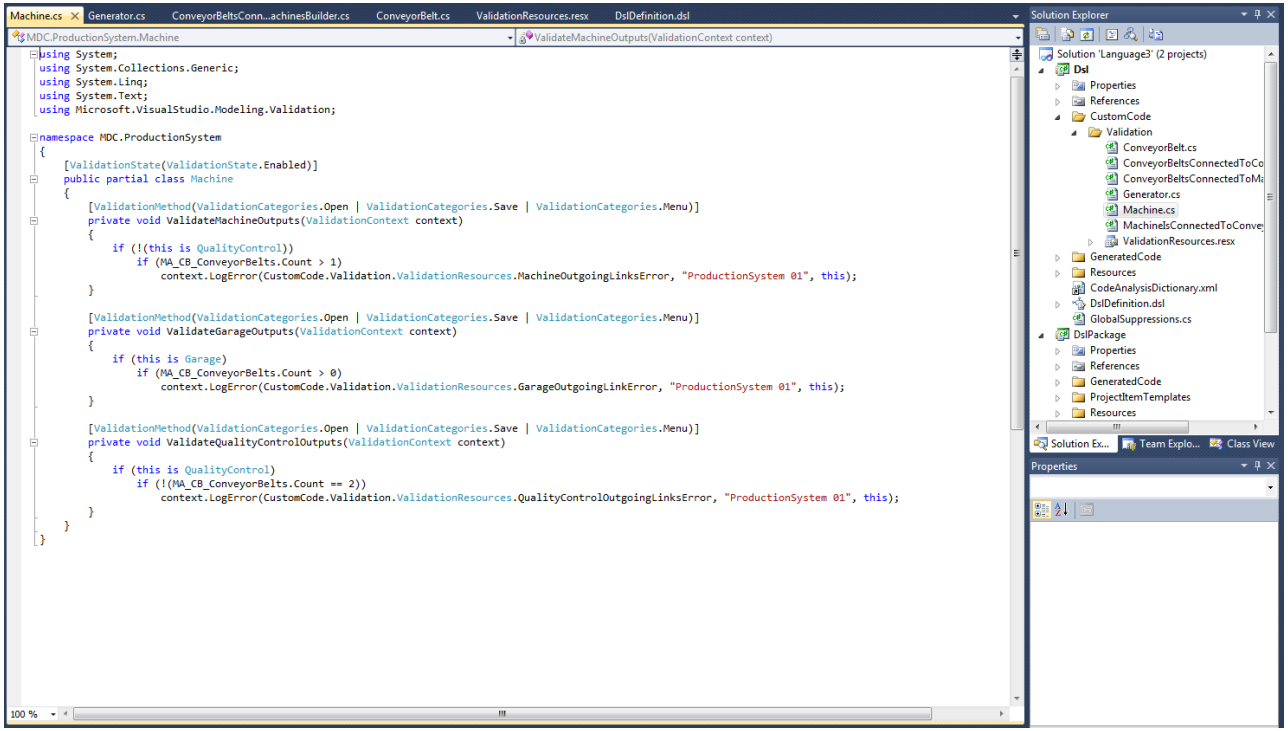


Validation

- Three types of validation
  - Validation on request
  - Interactive constraints
  - Hard constraints

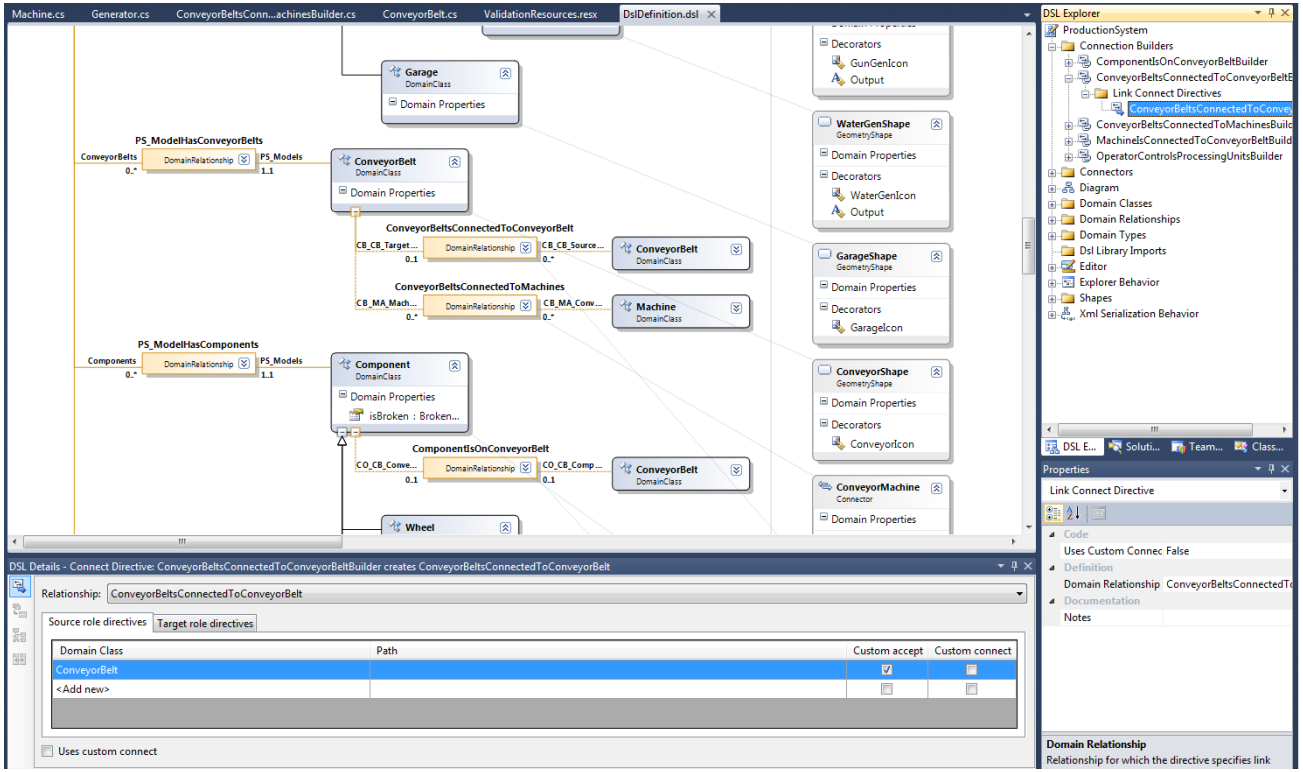
Validation

- Three types of validation
- Feels like a workaround

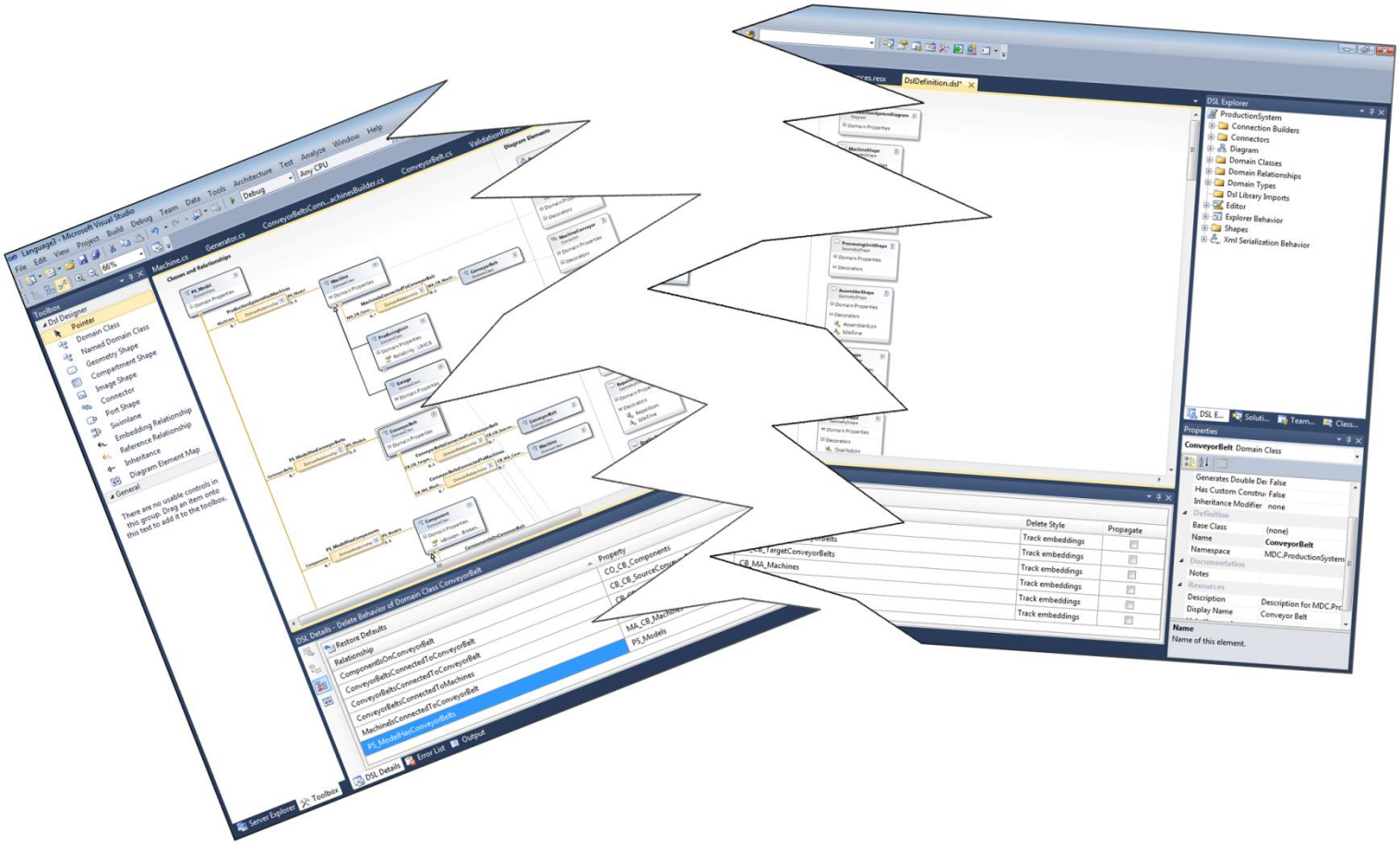


Validation

- Three types of validation
- Feels like a workaround



Simulation


DslDefinition.dsl*
DSL Explorer
ProductionSystem
Connection Builders
Connectors
Diagram
Domain Classes
Domain Relationships
Domain Types
Dsl Library Imports
Editor
Explorer Behavior
Shapes
Xml Serialization Behavior
Properties
ConveyorBelt Domain Class
Toolbox
Dsl Designer
Pointer
Domain Class
Named Domain Class
Geometry Shape
Compartment Shape
Image Shape
Connector
Port Shape
Swimlane
Embedding Relationship
Reference Relationship
Inheritance
Diagram Element Map
General
DSL Details - Delete Behavior of Domain Class ConveyorBelt
Restore Defaults
Relationship
ComponentIsOnConveyorBelt
ConveyorBeltsConnectedToConveyorBelt
ConveyorBeltsConnectedToMachines
MachinesConnectedToConveyorBelts
PS_ModelHasConveyorBelts
Property
CO_CB_Components
CB_CB_SourceConveyorBelts
CB_MA_Machines
MA_CB_Machines
PS_Models
Delete Style
Propagate
Track embeddings
Server Explorer
Toolbox
DSL Details
Error List
Output

Simulation – T4 Text Templates

```
<#@ template inherits="Microsoft.VisualStudio.TextTemplating.VSHost.ModelingTextTransformation" #>
<#@ output extension=".txt" #>
<#@ ProductionSystem processor="ProductionSystemDirectiveProcessor" requires="fileName='MachinesCantFire.pr'" #>

=======================================
Analysis of the file MachinesCantFire.pr
=======================================

The following is a list of all the machines currently operated by operators
also indicating if they are able to function or if there is an obstruction.
<#
  // When you change the DSL Definition, some of the code below may not work.

  foreach (Operator op in this.PS_Model.Operators)
  {
    if (op.OP_PU_ProcessingUnits != null)
    {
      Machine mach = op.OP_PU_ProcessingUnits;
      if (mach is RepairBay)
      {
        bool hasInput = false;
        bool canOutput = false;
        foreach (ConveyorBelt conv in mach.CB_MA_ConveyorBelts)
        {
          if (conv.CO_CB_Components != null)
          {
            hasInput = true;
          }
        }
        foreach (ConveyorBelt conv in mach.MA_CB_ConveyorBelts)
        {
          if (conv.CO_CB_Components == null)
          {
            canOutput = true;
          }
        }
        if (hasInput)
        {
          if (canOutput)
```

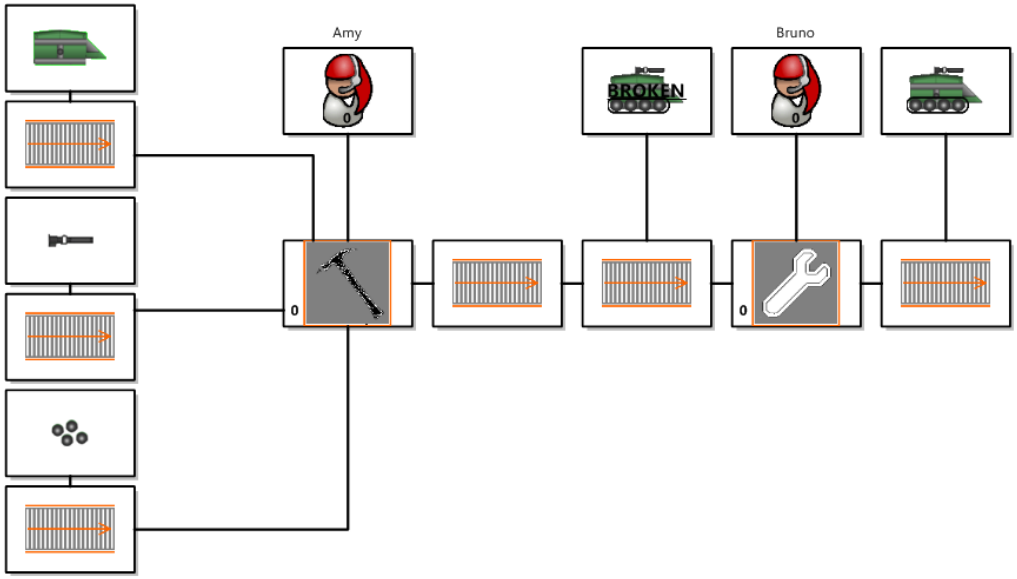
Simulation – T4 Text Templates

```
========================================
Analysis of the file MachinesCantFire.pr
========================================

The following is a list of all the machines currently operated by operators
also indicating if they are able to function or if there is an obstruction.

    The assembler operated by Amy is _NOT_ ready to produce,
    it requires additional components to begin production.

    The repair bay currently operated by Bruno is _NOT_ ready to repair the supplied component,
    either there is no conveyor belt leading out of this machine or there is already another item on it.
```



Comparison

- Strong points of VMSDK
  - Ease-of-use from a programmer's POV

Comparison

- Strong points of VMSDK
  - Ease-of-use from a programmer's POV
  - Everything under the same roof

Comparison

- Strong points of VMSDK
  - Ease-of-use from a programmer's POV
  - Everything under the same roof
  - Easy design of graphical object

Comparison

- Room for improvement
  - Support for simulation!

Comparison

- Room for improvement
  - Support for simulation!
  - Validation as a core concept

Comparison

- Room for improvement
  - Support for simulation!
  - Validation as a core concept
  - Automatic Link Detection

Comparison

- Room for improvement
  - Support for simulation!
  - Validation as a core concept
  - Automatic Link Detection
  - Learning Curve/Documentation

Comparison

- Room for improvement
  - Support for simulation!
  - Validation as a core concept
  - Automatic Link Detection
  - Learning Curve/Documentation
  - More options for cardinalities

Conclusion

- VMSDK is not a bad tool

Conclusion

- VMSDK is not a bad tool
- Less functionality than AToM3

Conclusion

- VMSDK is not a bad tool
- Less functionality than AToM3
- Step in the good direction

Conclusion

- VMSDK is not a bad tool
- Less functionality than AToM3
- Step in the good direction
- Use a good tutorial![3]

[3] Wills, A. C., June 2011.
Visualization and modeling sdk - intro lab.

Questions